AF/IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application No.: 10/722,614	§	
Filed: November 26, 2003	§	Examiner: Tsai, Sheng Jen
Inventor(s):	§	Group/Art Unit: 2186
Karr et al.	§	Atty. Dkt. No: 5760-09200
	§	
	§	I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Commissioner for Patents, Alexandria, VA 22313-1450, on the date indicated below.
Title: SYSTEM AND METHOD FOR EMULATING OPERATING SYSTEM METADATA TO PROVIDE CROSS-PLATFORM ACCESS TO STORAGE VOLUMES	§	B. Noël Kivlin Signature April 23, 2007 Date

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed on February 21, 2007, Appellants present this Appeal Brief. Appellants respectfully request that this appeal be considered by the Board of Patent Appeals and Interferences.

04/27/2007 RFEKADU1 00000028 501505 10722614
01 FC:1402 500.00 DA

I. REAL PARTY IN INTEREST

The present application is owned by VERITAS Operating Corporation, the assignee of record, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having an office and place of business at 350 Ellis Street, Mountain View, California 94043. VERITAS Operating Corporation is a subsidiary of Symantec Corporation, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 20330 Stevens Creek Boulevard, Cupertino, California 95014.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to Appellants, Appellants' legal representatives, or assignee which will directly affect, be directly affected by, or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1 – 28 are pending. Claims 1 – 28 are rejected, and the rejection of these claims is being appealed. A copy of claims 1 – 28 is included in the Claims Appendix attached hereto.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed towards a storage subsystem comprising at least one storage device (*see*, e.g., Figs. 1, 3, and 4, reference characters 104A, 104B, 104C; page 9, lines 15 – 30; page 10, lines 11 – 16) and a storage virtualization controller (*see*, e.g., Fig. 3, reference character 102; page 13, line 17 to page 14, line 5; page 14, lines 15 – 21). The storage virtualization controller is communicatively coupled to the at least one storage device (*see*, e.g., Fig. 3, reference character 256; page 14, lines 15 – 19). The storage virtualization controller is operable to determine a metadata format usable to access data stored on the at least one storage device under a first operating system (*see*, e.g., Fig. 3, reference character 130; page 15, lines 11 – 19). The metadata format is determined in response to a request by a host computer system to access the data (*see*, e.g., Fig. 3, reference character 106; page 14, lines 10 – 13; page 18, lines 7 – 20), and the metadata format is determined based on the host computer system running the first operating system (*see*, e.g., Fig. 3, reference characters 106, 130; page 15, lines 21 – 30; page 16, lines 22 – 27). The storage virtualization controller is also operable to generate operating system metadata in accordance with the determined metadata format for the at least one storage device (*see*, e.g., Fig. 5, reference character 301; page 16, line 22 to page 17, line 4). The operating system metadata emulates a storage volume hosted under the first operating system (*see*, e.g., Fig. 4, reference character 107; Fig. 5, reference character 301; page 15, lines 21 – 30; page 17, lines 6 – 25; page 18, line 22 to page 20 line 2). The storage virtualization controller is further operable to send the operating system metadata to the host computer system (*see*, e.g., Fig. 5, reference character 303; page 17, lines 6 – 10). The operating system metadata enables the host computer system to recognize the storage device as the storage volume hosted under the first operating system (*see*, e.g., Fig. 4, reference character 107; Fig. 5, reference character 303; page 15, lines 21 – 30; page 17, lines 6 – 25).

Independent claim 14 is directed towards a method that comprises determining a metadata format usable to access data stored on a storage device (*see*, e.g., Figs. 1, 3, and 4, reference characters 104A, 104B, 104C; page 9, lines 15 – 30; page 10, lines 11 – 16)

under a first operating system (*see*, e.g., Fig. 3, reference character 130; page 15, lines 11 – 19), wherein the metadata format is determined in response to a request by a host computer system to access the data (*see*, e.g., Fig. 3, reference character 106; page 14, lines 10 – 13; page 18, lines 7 – 20), and wherein the metadata format is determined based on the host computer system running the first operating system (*see*, e.g., Fig. 3, reference characters 106, 130; page 15, lines 21 – 30; page 16, lines 22 – 27). The method also comprises generating operating system metadata in accordance with the determined metadata format for the storage device (*see*, e.g., Fig. 5, reference character 301; page 16, line 22 to page 17, line 4), wherein the operating system metadata emulates a storage volume hosted under the first operating system (*see*, e.g., Fig. 4, reference character 107; Fig. 5, reference character 301; page 15, lines 21 – 30; page 17, lines 6 – 25; page 18, line 22 to page 20 line 2). The method further comprises sending the operating system metadata to the host computer system (*see*, e.g., Fig. 5, reference character 303; page 17, lines 6 – 10), wherein the operating system metadata enables the host computer system to recognize the storage device as the storage volume hosted under the first operating system (*see*, e.g., Fig. 4, reference character 107; Fig. 5, reference character 303; page 15, lines 21 – 30; page 17, lines 6 – 25).

Independent claim 27 is directed towards a computer-readable storage medium comprising program instructions (*see*, e.g., Fig. 2, reference characters 230A, 240; Fig. 3, reference character 230B; page 12, line 19 to page 13, line 2; page 14, lines 15 – 17; page 22, line 25 to page 23, line 2). The program instructions are computer-executable to implement determining a metadata format usable to access data stored on a storage device (*see*, e.g., Figs. 1, 3, and 4, reference characters 104A, 104B, 104C; page 9, lines 15 – 30; page 10, lines 11 – 16) under a first operating system (*see*, e.g., Fig. 3, reference character 130; page 15, lines 11 – 19), wherein the metadata format is determined in response to a request by a host computer system to access the data (*see*, e.g., Fig. 3, reference character 106; page 14, lines 10 – 13; page 18, lines 7 – 20), and wherein the metadata format is determined based on the host computer system running the first operating system (*see*, e.g., Fig. 3, reference characters 106, 130; page 15, lines 21 – 30; page 16, lines 22 – 27). The program instructions are also computer-executable to

implement generating operating system metadata in accordance with the determined metadata format for the storage device (*see*, e.g., Fig. 5, reference character 301; page 16, line 22 to page 17, line 4), wherein the operating system metadata emulates a storage volume hosted under the first operating system (*see*, e.g., Fig. 4, reference character 107; Fig. 5, reference character 301; page 15, lines 21 – 30; page 17, lines 6 – 25; page 18, line 22 to page 20 line 2). The program instructions are further computer-executable to implement sending the operating system metadata to the host computer system (*see*, e.g., Fig. 5, reference character 303; page 17, lines 6 – 10), wherein the operating system metadata enables the host computer system to recognize the storage device as the storage volume hosted under the first operating system (*see*, e.g., Fig. 4, reference character 107; Fig. 5, reference character 303; page 15, lines 21 – 30; page 17, lines 6 – 25).

Independent claim 28 is directed towards a system comprising means for determining a metadata format usable to access data stored on a storage device (*see*, e.g., Figs. 1, 3, and 4, reference characters 104A, 104B, 104C; Fig. 3, reference character 102; page 9, lines 15 – 30; page 10, lines 11 – 16; page 13, line 17 to page 14, line 5; page 14, lines 15 – 21) under a first operating system (*see*, e.g., Fig. 3, reference character 130; page 15, lines 11 – 19), wherein the metadata format is determined in response to a request by a host computer system to access the data (*see*, e.g., Fig. 3, reference character 106; page 14, lines 10 – 13 and 23 – 27; page 18, lines 7 – 20), and wherein the metadata format is determined based on the host computer system running the first operating system (*see*, e.g., Fig. 3, reference characters 106, 130; page 15, lines 21 – 30; page 16, lines 22 – 27). The system also comprises means for generating operating system metadata in accordance with the determined metadata format for the storage device (*see*, e.g., Fig. 3, reference character 102; Fig. 5, reference character 301; page 13, line 17 to page 14, line 5; page 14, lines 15 – 21; page 15, lines 21 – 30; page 16, line 22 to page 17, line 4), wherein the operating system metadata emulates a storage volume hosted under the first operating system (*see*, e.g., Fig. 3, reference character 130; Fig. 4, reference character 107; Fig. 5, reference character 301; page 15, lines 21 – 30; page 17, lines 6 – 25; page 18, line 22 to page 20 line 2). The system further comprises means for sending the operating system metadata to the host computer system (*see*, e.g., Fig. 3,

reference characters 102, 106; Fig. 5, reference character 303; page 13, line 17 to page 14, line 5; page 14, lines 15 – 27; page 17, lines 6 – 10), wherein the operating system metadata enables the host computer system to recognize the storage device as the storage volume hosted under the first operating system (*see*, e.g., Fig. 3, reference characters 108, 130, 250; Fig. 4, reference character 107; Fig. 5, reference character 303; page 15, lines 21 – 30; page 17, lines 6 – 25).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1 – 28 stand rejected under 35 U.S.C. §102(e) as being anticipated by Rajan et al. (U.S. Publication No. US 2004/0030822, hereinafter "Rajan").

VII. ARGUMENT

First Ground of Rejection:

Claims 1 – 28 stand rejected under 35 U.S.C. §102(e) as being anticipated by Rajan et al. (U.S. Publication No. US 2004/0030822, hereinafter "Rajan"). Appellants traverse this rejection for the following reasons.

Claims 1 – 6, 8 – 19, and 21 – 28:

Appellants respectfully submit that Rajan does not teach or suggest a storage virtualization controller operable to generate operating system metadata in accordance with the determined metadata format for the at least one storage device, wherein the metadata format is determined in response to a request by a host computer system to access the data, and wherein the operating system metadata enables the host computer system to recognize the storage device as the storage volume hosted under the first operating system, in combination with the remaining features of claim 1. Thus, in Appellants' claim 1, the operating system metadata is generated in accordance with the

determined metadata format for the at least one storage device in response to a request by a host computer system to access the data.

Rajan discloses a storage appliance which pools storage to create "vdisks" of varying size in response to user requests. The storage appliance provides access to the vdisks for various different storage configurations (e.g., NAS and SAN) and corresponding access protocols (e.g., CIFS, NFS, FibreChannel, etc.). However, the metadata format for the data on the storage appliance is fixed according to the requirements of the storage operating system installed on the storage appliance (see, e.g., Figures 3 and 4 and sections [0034], [0035], [0044], and [0049]). The operating system metadata in Rajan's vdisks is initially generated when the storage operating system is installed by user request, i.e., before any request by a host computer system to access the data. Therefore, Rajan does not teach or suggest a storage virtualization controller operable to generate operating system metadata in accordance with a metadata format determined in response to a request by a host computer system to access the data.

Anticipation requires the presence of each and every limitation of the claimed invention, arranged as in the claim, in a single prior art reference. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Rajan fails to disclose a storage subsystem comprising a storage virtualization controller, wherein the storage virtualization controller is operable to "generate operating system metadata in accordance with the determined metadata format for the at least one storage device," "wherein the metadata format is determined in response to a request by a host computer system to access the data," and "wherein the operating system metadata enables the host computer system to recognize the storage device as the storage volume hosted under the first operating system," in combination with the remaining features of claim 1. Therefore, Rajan cannot be said to anticipate claim 1.

Accordingly, claim 1 and its dependent claims 2 – 6 and 8 – 13 are believed to patentably distinguish over Rajan for at least the reasons given above. Claims 14, 27, and 28 include features similar to those of claim 1 and are believed to patentably distinguish over Rajan for at least the same reasons. Dependent claims 15 – 19 and 21 – 26 are also believed to patentably distinguish over Rajan for similar reasons.

Claims 7 and 20:

Claim 7 depends on claim 1 and is therefore believed to patentably distinguish over the art cited by the Final Office Action for the reasons given above with respect to claim 1. Additionally, Appellants respectfully submit that Rajan does not teach or suggest a storage subsystem "wherein in generating the operating system metadata for the storage device, the storage virtualization controller is operable to add a storage property to identify an offset and a length of the storage volume" in combination with the remaining features of claims 1 and 7. In rejecting claim 7, the Final Office Action cites Fig. 4 of Rajan. Fig. 4 illustrates various elements (e.g., a size) in an instance of metadata for one file in a file system (see, e.g., section [0044]), not a storage property identifying an offset and a length of a storage volume. The Final Office Action also cites Fig. 3 of Rajan in rejecting claim 7. Fig. 3 illustrates various elements of metadata associated with a vdisk, but not a storage property identifying an offset and a length of a storage volume. There is no teaching or suggestion in Rajan for adding a storage property to identify an offset and a length of the storage volume in generating the operating system metadata for the storage device.

Accordingly, claim 7 is believed to patentably distinguish over Rajan for at least the reasons given above. Claim 20 includes features similar to those of claim 7 and is believed to patentably distinguish over Rajan for at least the same reasons.

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1 – 28 was erroneous, and reversal of the decision is respectfully requested.

VIII. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A storage subsystem, comprising:

at least one storage device; and

a storage virtualization controller, wherein the storage virtualization controller is communicatively coupled to the at least one storage device, and wherein the storage virtualization controller is operable to:

determine a metadata format usable to access data stored on the at least one storage device under a first operating system, wherein the metadata format is determined in response to a request by a host computer system to access the data, and wherein the metadata format is determined based on the host computer system running the first operating system;

generate operating system metadata in accordance with the determined metadata format for the at least one storage device, wherein the operating system metadata emulates a storage volume hosted under the first operating system; and

send the operating system metadata to the host computer system, wherein the operating system metadata enables the host computer system to recognize the storage device as the storage volume hosted under the first operating system.

2. The storage subsystem of claim 1,

wherein the operating system metadata enables a block storage I/O stack in the first operating system on the host computer system to recognize the storage device as a partition.

3. The storage subsystem of claim 1,

wherein the operating system metadata enables a block storage I/O stack in the first operating system on the host computer system to recognize the storage device as a host-virtual object.

4. The storage subsystem of claim 1,

wherein the operating system metadata enables a driver on the host computer system to recognize the storage device as an enclosed volume, wherein the driver is layered above a block storage I/O stack in the first operating system.

5. The storage subsystem of claim 1,

wherein the storage virtualization controller is operable to configure the operating system metadata in response to a requirement of the first operating system.

6. The storage subsystem of claim 1,

wherein a management environment is configured to supply operating system types and operating system metadata configuration requirements to the storage virtualization controller, wherein the operating system types comprise the first operating system.

7. The storage subsystem of claim 1,

wherein in generating the operating system metadata for the storage device, the storage virtualization controller is operable to add a storage property to identify an offset and a length of the storage volume.

8. The storage subsystem of claim 1,

wherein an operation is provided to configure operating system types and operating system metadata configuration requirements for generating the operating system metadata, wherein the operating system types comprise the first operating system.

9. The storage subsystem of claim 1,

wherein the storage virtualization controller is operable to receive user input to select one of a plurality of operating system types for the operating system metadata, wherein the operating system types comprise the first operating system.

10. The storage subsystem of claim 1,

wherein the storage virtualization controller is operable to send an operating system metadata configuration instruction to the storage device through a vendor-unique I/O request to the storage device.

11. The storage subsystem of claim 1,

wherein the operating system metadata emulates a storage volume hosted under a first operating system and one or more additional operating systems; and

wherein the operating system metadata enables a layered driver on the host computer system to recognize the storage device.

12. The storage subsystem of claim 1,

using a layered driver on the host computer system to provide access to a storage volume mapped within a Logical Unit, wherein the Logical Unit is provided by an external device or an external virtualization layer.

13. The storage subsystem of claim 1,

wherein a management environment is configured to supply a preferred name of the storage device to software on the host computer system.

14. A method comprising:

determining a metadata format usable to access data stored on a storage device under a first operating system, wherein the metadata format is determined in response to a request by a host computer system to access the data, and wherein the metadata format is determined based on the host computer system running the first operating system;

generating operating system metadata in accordance with the determined metadata format for the storage device, wherein the operating system metadata emulates a storage volume hosted under the first operating system; and

sending the operating system metadata to the host computer system, wherein the operating system metadata enables the host computer system to recognize the storage device as the storage volume hosted under the first operating system.

15. The method of claim 14,

wherein the operating system metadata enables a block storage I/O stack in the first operating system on the host computer system to recognize the storage device as a partition.

16. The method of claim 14,

wherein the operating system metadata enables a block storage I/O stack in the first operating system on the host computer system to recognize the storage device as a host-virtual object.

17. The method of claim 14,

wherein the operating system metadata enables a driver on the host computer system to recognize the storage device as an enclosed volume, wherein the driver is layered above a block storage I/O stack in the first operating system.

18. The method of claim 14, further comprising:

configuring the generating the operating system metadata in response to a requirement of the first operating system.

19. The method of claim 14,

wherein the generating the operating system metadata for the storage device is performed by a storage virtualizer; and

wherein a management environment is configured to supply operating system types and operating system metadata configuration requirements to the storage virtualizer, wherein the operating system types comprise the first operating system.

20. The method of claim 14,

wherein the generating the operating system metadata for the storage device comprises adding a storage property to identify an offset and a length of the storage volume.

21. The method of claim 14,

wherein an operation is provided to configure operating system types and operating system metadata configuration requirements for the generating the operating system metadata, wherein the operating system types comprise the first operating system.

22. The method of claim 14, further comprising:

receiving user input to select one of a plurality of operating system types for the operating system metadata, wherein the operating system types comprise the first operating system.

23. The method of claim 14, further comprising:

sending an operating system metadata configuration instruction to the storage device through a vendor-unique I/O request to the storage device.

24. The method of claim 14,

wherein the operating system metadata emulates a storage volume hosted under a first operating system and one or more additional operating systems; and

wherein the operating system metadata enables a layered driver on the host computer system to recognize the storage device.

25. The method of claim 14,

using a layered driver on the host computer system to provide access to a storage volume mapped within a Logical Unit, wherein the Logical Unit is provided by an external device or an external virtualization layer.

26. The method of claim 14,

wherein a management environment is configured to supply a preferred name of the storage device to software on the host computer system.

27. A computer-readable storage medium comprising program instructions, wherein the program instructions are computer-executable to implement:

determining a metadata format usable to access data stored on a storage device under a first operating system, wherein the metadata format is determined in response to a request by a host computer system to access the data, and wherein the metadata format is determined based on the host computer system running the first operating system;

generating operating system metadata in accordance with the determined metadata format for the storage device, wherein the operating system metadata emulates a storage volume hosted under the first operating system; and

sending the operating system metadata to the host computer system, wherein the operating system metadata enables the host computer system to recognize the storage device as the storage volume hosted under the first operating system.

28. A system comprising:

means for determining a metadata format usable to access data stored on a storage device under a first operating system, wherein the metadata format is determined in response to a request by a host computer system to access the data, and wherein the metadata format is determined based on the host computer system running the first operating system;

means for generating operating system metadata in accordance with the determined metadata format for the storage device, wherein the operating system metadata emulates a storage volume hosted under the first operating system; and

means for sending the operating system metadata to the host computer system, wherein the operating system metadata enables the host computer system to recognize the storage device as the storage volume hosted under the first operating system.

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131, or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings known to Appellants, Appellants' legal representatives, or assignee which will directly affect, be directly affected by, or have a bearing on the Board's decision in the pending appeal.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5760-09200/BNK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

B. Noël Kivlin
Reg. No. 33,929
ATTORNEY FOR APPLICANTS

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: April 23, 2007